Math 228B
Class Notes for week 6, lecture 11

Non-rectangular Domain
We want to solve the eqution

$$u_t = b\triangle u$$

in a non-rectangular domain with Dirichlet BCs.
There are few options:

1. use a Cartesian grid
2. use body fitted coordinates (see Polar Coordinates example below)
3. use unstructureed mesh (note that this method works better for finite volume and finite element)

- Cartesian grid

  For example, Ω is a square with a circle removed, then there are two boundaries, namely the outter square and the inner circle. Let's define the points for this discretization.

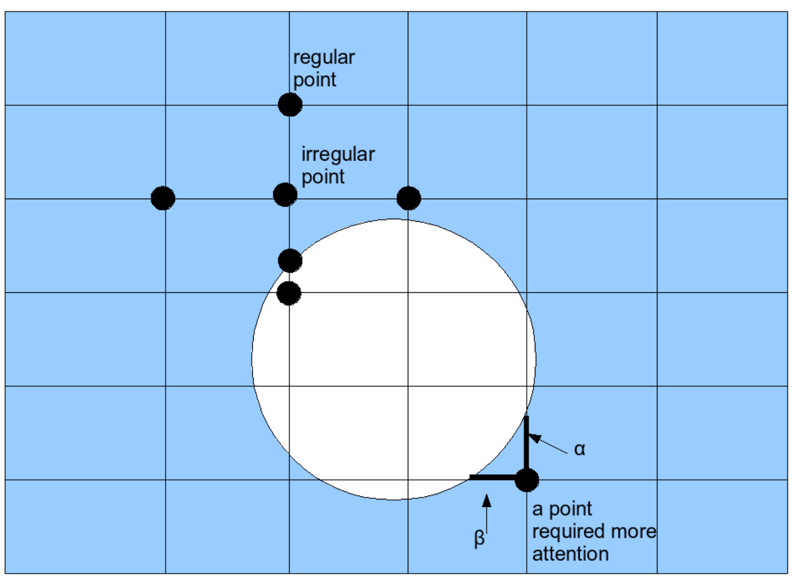
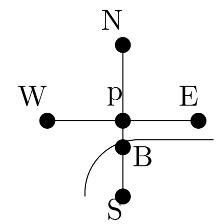


Figure 1: Ω is square with circle removed.

We will denote regular points as points have only neighbors inside the domain, and irregular



points with at least one neighbor located outside the domain.

How to discretize at point p (irregular point)?
Let denote the Laplacian operator

$$L = L_x + L_y,$$

where L_x uses standard discretization (W,p,E are inside the domain). For L_y,

– use N, p, B to extrapolate a value to s, and apply the standard stencil. However, you can run into trouble applying this method (Figure 1), as graph shows, require two directions.

It can be shown, using the first method, suppose that $y_p - y_B = \alpha h$ for $h < 1$, we have

$$u_{xx} \approx \frac{2\alpha u_N - 2(\alpha+1)u_p + 2u_B}{\alpha(\alpha)h^2},$$

where α is the distance between point p and point B.

- Implementation
How to invert the operator? and, How to handle storage?
Need to store the coefficients of the discrete Laplacian at each point.

  What about points outside the domain?
  Can use the regular Laplacian operator.
  Can use the identity operator (they decouple).

  How to solve?
  1. Use iterative methods
  –SOR, just need to change the stencil accordingly.
  –MG, need nontrivial change to existing code.

  2. For time-dependent equations (e.g., the heat equation)
  –ADI, need to be careful with forming the tri-diagonal matrices in your solver.

  What to program?
  decide if a point is in or out, and then change the neighbors, if points are irregular, then you need to calculate the distances (α and β) to the boundary.

- Stability
Want to avoid explicit schemes, even if $\frac{h^2}{4b}$ is not small. For stability, need $\triangle t \leq \frac{h^2}{2b(\frac{1}{\alpha}+\frac{1}{\beta})}$, we can see that the restriction is placed on α and β.

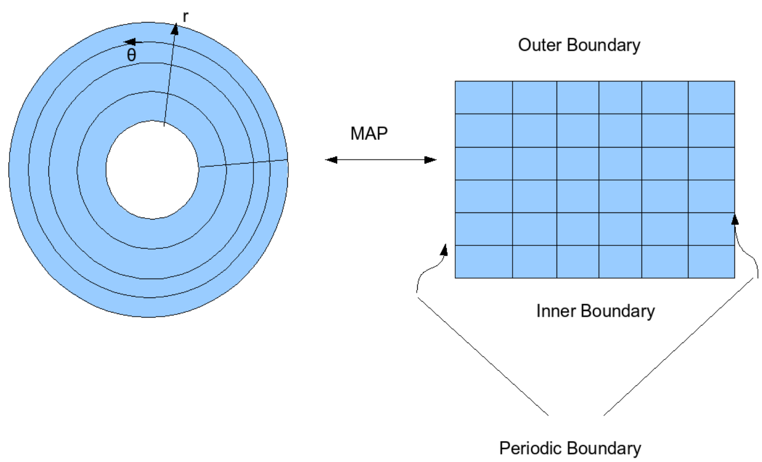


Figure 2: Mapping of Polar and standard coordinates. Given boundary conditions on top and bottom, periodic boundary conditions on left and right.

- Accuracy
  At the irregular points, the LTE is of $O(h)$. In one dimension (see 228A HW), on $O(h)$ LTE adjacent to the boundary contributes an $O(h^3)$ error to the solution. This holds in higher dimensions, but it's much more difficult to show.

- Boundary fitted meshes
  Idea: map the physical domain to rectangular domain and solve on the transformed domain. An easy example: Polar Coordinates

  Regular mesh for r and θ are transformed by

$$x = r\cos\theta$$
$$y = r\sin\theta$$

  We also need to transform the PDE.

$$\triangle u = \frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial u}{\partial r}) + \frac{1}{r}\frac{\partial^2 u}{\partial \theta^2}.$$

  Note that now the PDE is a variable coefficient equation (see Figure 2).